Control Techniques and Programming Issues for Time Delayed Internet Based Teleoperation

Saghir Munir and Wayne J. Book

George W.Woodruff School of Mechanical Engineering
Georgia Institute Of Technology
Atlanta, GA, 30332, U.S.A
saghir.munir@intel.com
wayne.book@me.gatech.edu
http://www.imdl.gatech.edu

Abstract— **This article focuses on Internet-based real time control, such as remote bilateral teleoperation. In such applications it is required that the control loop be closed around a time delayed network. Although various researchers have worked on this problem, this paper focuses on two control strategies (based on wave variables and a time forward observer), bandwidth issues, and some related programming details. Experimental results of bilateral teleoperation via the Ethernet between Atlanta and Tokyo are given. The system used was a two degree of freedom haptic interface, bilaterally coupled to simulation (implemented on a windows NT based computer) of a similar system.**

I. Introduction

In the past 8 years, numerous researchers have been trying to incorporate the Internet into the control of robots. The Internet, which was once used for simple data transfer, has now evolved to be used to electronically communicate, pay bills, learn, explore, trade stocks in real time and now 'control' physical hardware in real time.

In 1999 the IEEE Robotics & Automation magazine devoted the September issue to 'Robots on the Web'. In that issue Taylor and Dalton [18] report on the evolution of Internet robotics and on their work at the University of Western Australia where a six axis robot has been made teleoperable over the Internet since 1994. Control is established via a JAVA based applet running in a web browser. The user sends reference points over the browser interface and the control loop, which is local to the robot, executes the reference command and notifies the user when the task is complete. The user gets no force feedback and has no control over the process during the time a task is being performed.

A case where the control signal (and not the reference signal) flows over the Internet has been reported by Overstreet and Tzes in [15]. In this configuration, the loop is closed around the network. Their work provides remote-access to laboratory facilities (robotic equipment) from virtually anywhere. Their Internet-accessed remote laboratory is based on a client/server computer configuration. The server, situated near the experiment receives command signals transmitted by the client. Issues concerning network reliability, dynamic delays caused by Internet traffic, concurrent user access and limited computing power were briefly addressed. However the paper did not highlight on instabilities caused by the added lag introduced in the control loop by the Internet, nor did the researchers try to compensate for the delay in any form. The idea of extending the use of laboratory equipment to students over the Ethernet has recently caught the attention of several academic institutions, as it provides easy and convenient access.

The control community has seen a lot of literature related to network-based teleoperation. However an in-depth analysis of the nature of time delay over very large distances has to the best of our knowledge only been attempted in the early days of the net by Oboe and Fiorini [14] of the Jet Propulsion Laboratory. Although their paper was comprehensive at the time, we feel the network has evolved since then, hence its characteristics need to be re-examined with particular attention to bandwidth issues (and related packet loss) which ultimately addresses the frequency and the reliability with which the control loop can be executed.

Among the first work ever published tackling time delay over the Internet for bilateral teleoperation in a rigorous fashion was by Tran [17] and Brady([3] and [4]). From a theoretical perspective this was based on earlier work of Watanabe [19], which relied on the use of time forward observer. A few years later Slotine and Niemeyer ([9], [10], [11] [12] and [13]) presented another formulation tackling the same problem using wave variables. This evolved from the earlier work of Anderson and Spong ([1] and [2]). Recently wave-based formulation was enhanced using a prediction technique to improve performance (especially the settling time) by the authors of this paper([6], [7] and [8]).

In this research existing techniques based on passivity (wave variables) and scattering theory to establish Internet based teleoperation with prediction, for variable delay are briefly summarized[1]. Emphasis is given to the experimental implementation of this control strategy and the resulting system performance. In conventional wave-based techniques stability comes at the expense of degrading system performance with increasing time delay. In order to enhance performance, a prediction technique is employed to cut down the lag felt by the user, while simultaneously

[1]The reader is referred to [6] and [7] for a thorough background.

improving the settling time. The underlying controller is shown to be robustly stable to model mismatches since passivity is explicitly enforced by an energy regulator. It is also shown that even for a mismatching plant model (in the predictor), the forces felt by the master and slave at steady state are identical.

In an alternative scheme the use of a time forward observer derived from conventional state-space analysis to ensure stability under a varying delay is made. In essence this is almost similar to the work of Tarn and Brady, however it was felt that a complete and rigorous proof of this formalism was missing from literature. It is shown that the observer is easy to implement and has performance characteristics superior to passivity based formalisms. However the observer is not robust to model mismatches and is unable to reflect remote forces back to the master at steady state conditions while in hard contact. This scheme is primarily geared towards stabilizing a time delayed system, and not force reflection.

The two main ideas developed in this article were experimentally validated on a 2-degrees of freedom bilateral teleoperator. The loop was closed over the Internet, with the closed-loop distance comparable to the circumference of Earth. The issue of bandwidth is examined, particularly where a large number of states are streamed through the network at a high frequency. It was seen that very large packet loss occurs if the time step is made sufficiently small. Further, details are given on how to use Window NT (an 'event-based' operating system) to implement a simulation in real time.

Section 2 focuses on familiarizing the reader with a basic bilateral teleoperator, followed by section 3 which includes a brief summary of wave variables. Section 4 focuses on a time forward observer to stabilize a system with varying time delay. In section 5, the nature of the Internet time delay is discussed, along with the experimental setup. Section 6 focuses on experimental results, and finally some concluding remarks are given in section 7.

II. Bilateral Teleoperator

Figure 1 shows the arrangement of a bilateral teleoperator. Both the master and the slave robots are feedback linearized to yield a linear system. In this case the equation of motion for the master (and slave) manipulator is given by

$$v_m = D_m(\theta_m)\ddot{\theta}_m + h_m(\theta_m, \dot{\theta}_m) + N_m(\theta_m) \qquad (1)$$

where v_m is the input torque, $D_m(\theta_m)$ is a positive definite symmetric inertia matrix (with its rank being equal to the degrees of freedom of the system), while $h_m(\theta_m, \dot{\theta}_m)$ and $N_m(\theta_m)$ both have non linear elements. It is desired that the feedback linearized system be of the form

$$\tau_{pd} = J_m\ddot{\theta}_m + B_m\dot{\theta}_m \qquad (2)$$

where τ_{pd} is the input torque, J_m is the desired constant positive definite symmetric inertia matrix and B_m is the

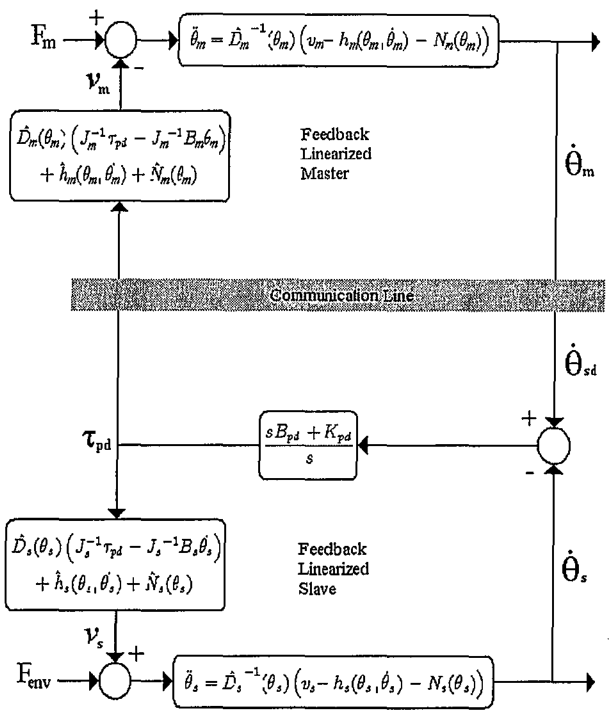

Fig. 1. A multi-degree of freedom bilateral teleoperator.

damping matrix (also positive definite, symmetric and constant). We choose a controller such that the input torque v_m, to the slave system is computed according to

$$\begin{aligned} v_m = \hat{D}_m(\theta_m)\left(J_m^{-1}\tau_{pd} - J_m{}^{-1}B_m\dot{\theta}_m\right) \\ +\hat{h}_m(\theta_m, \dot{\theta}_m) + \hat{N}_m(\theta_m) \end{aligned} \qquad (3)$$

where the parameters with the *hat* are estimates of the actual system. Setting equation (3) equal to equation (2) and assuming that there is no modeling error, yields the desired linear dynamics given by equation (2). This scheme is indicated in figure 1. The output of the PD controller is given by

$$\tau_{pd} = K_{pd}(\theta_{sd} - \theta_s) + B_{pd}(\dot{\theta}_{sd} - \dot{\theta}_s). \qquad (4)$$

III. Wave Variables

The above system performs well for as long as there is no delay in the loop. However as the delay increases, performance starts to degrade and the system very quickly becomes unstable. This problem is attributed to the non-passive nature of the communication link, where it can be seen that torque and velocity have multiplicative dependence on the instantaneous power-input (to the communication line) defined as

$$P_{in} = \dot{\theta}_m^T\tau_m - \dot{\theta}_{sd}^T\tau_{pd}. \qquad (5)$$

This dependance can be eliminated using the wave transformations

$$\begin{aligned} u_m(t) = A_w\dot{\theta}_m(t) + B_w\tau_m(t) \\ v_s(t) = C_w\dot{\theta}_{sd}(t) - D_w\tau_{pd}(t) \end{aligned} \qquad (6)$$

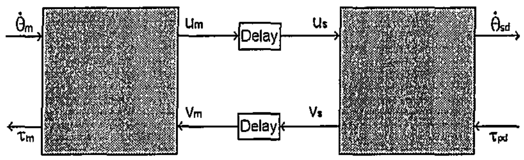

Fig. 2. Wave-based communication, by transforming velocity-force variables to wave variables before transmission and then back to velocity-force variable after transmission. Note: The gray boxes indicate wave transformations.

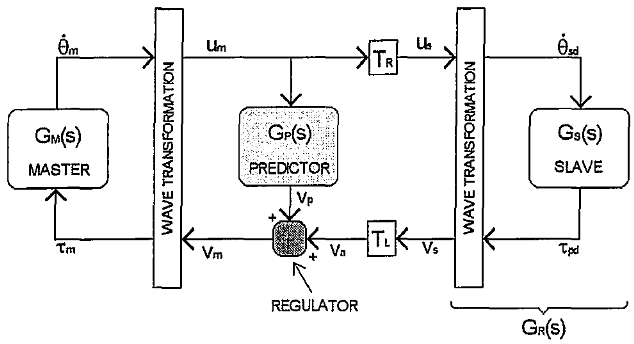

Fig. 3. A possible arrangement of a predictor incorporated inside the wave junction.

and

$$\begin{aligned} v_m(t) &= C_w\dot{\theta}_m(t) - D_w\tau_m(t) \\ u_s(t) &= A_w\dot{\theta}_{sd}(t) + B_w\tau_{pd}(t) \end{aligned} \tag{7}$$

where A_w, B_w, C_w and D_w are $n \times n$ scaling matrices (n being the number of degrees of freedom of the teleoperator) which satisfy[2] the following equations

$$\begin{aligned} A_w &= C_w \\ B_w &= D_w \\ I &= 2A_wB_w. \end{aligned} \tag{8}$$

Here wave variables (u and v), rather than power variables (τ and θ), are transmitted across the communication line (i.e. the communication line is replaced with that shown in figure 2). Using the above wave transformations it can be shown that the net power-in (equation 5) no longer has a multiplicative dependence on torque and velocity, but rather has a additive dependence. Hence when signals are temporarily delayed in the communication link, the line does not appear to be non-passive.

A. *Wave Prediction*

A possible arrangement of a Smith-type [16] wave predictor is shown in figure 3. Here $G_M(s)$ is the transfer function of the master manipulator, $G_S(s)$ is the transfer function of the slave and PD controller combined, and $G_P(s)$ is the transfer function of the predictor. The two rectangular boxes represent wave transformations and $G_R(s)$ is the combined transfer function of the entire right hand side (i.e. the slave, the PD controller and the wave transformation). T_R and T_L represent the time delays in the right and left directions respectively. The box marked as the *REGULATOR* for now can be assumed to be a summing junction.

[2]The reader is referred to the author's Ph.D thesis [6] for a more detailed proof of this analysis.

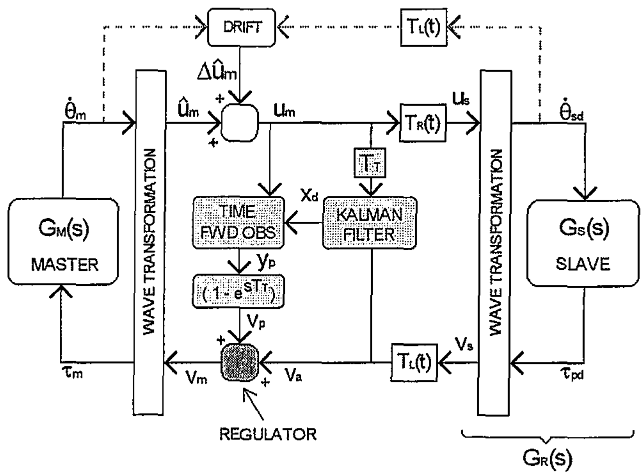

Fig. 4. Overall prediction scheme.

It can be shown that position information is encoded in the integral of of the wave signals u and v. Hence if the predictor is not designed with care, these integrals might not be preserved, leading to a non-zero steady state error. Writing out an expression for position difference across the wave junction yields

$$\Delta\theta(t) = \theta_m(t) - \theta_{sd}(t) \tag{9}$$

which, written in terms of wave variables becomes

$$\Delta\theta(t) = \frac{1}{2}{A_w}^{-1}\int_0^t (u_m + v_m - u_s - v_s)\ d\tau. \tag{10}$$

Taking the Laplace transform of this and expanding, yields

$$\begin{aligned} \Delta\Theta(s) = \tfrac{1}{2s}{A_w}^{-1}\left(1 - e^{-sT_R} + G_P(s)\right)U_m(s) \\ -\tfrac{1}{2s}{A_w}^{-1}\left(1 - e^{sT_L}\right)V_s(s). \end{aligned} \tag{11}$$

For zero steady state error

$$\lim_{t\to\infty}\Delta\theta(t) = \lim_{s\to 0} s\Delta\Theta(s) = 0. \tag{12}$$

Since at steady state the wave signals decay to zero, it is required that

$$\lim_{s\to 0} G_P(s) = 0. \tag{13}$$

In order to ensure passivity, the predictor must not increase the total return energy of the system. Hence it is required that

$$\int_0^t \frac{1}{2}{v_a}^T v_a\ d\tau \geq \int_0^t \frac{1}{2}{v_m}^T v_m\ d\tau. \tag{14}$$

This condition is explicitly enforced by the *REGULATOR* (discussed later in this section). Figure 4 shows a possible arrangement of a predictor. In this arrangement a Kalman filter first estimates the internal state of the plant, which is delayed by an amount T_T, where

$$T_T = T_R + T_L. \tag{15}$$

The time forward observer then uses the output of the Kalman filter to march the state T_T seconds into the future and computes the output y_p, which is then used to obtain a prediction signal v_p according to

$$V_p(s) = (1 - e^{sT_T})Y(s). \tag{16}$$

Notice that this expression satisfies the tracking condition given by equation (13).

B. Predictor implementation

The entire right hand side of the system (marked $G_R(s)$ in figure 4) can be represented as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu_m(t - T_R) \\ v_s(t) &= Cx(t) + Du_m(t - T_R). \end{aligned} \tag{17}$$

Using the variable $x_d(t) = x(t - T_L)$, the above expression can be written as

$$\begin{aligned} \dot{x}_d(t) &= Ax_d(t) + Bu_m(t - T_T) \\ v_a(t) &= v_s(t - T_L) = Cx_d(t) + Du_m(t - T_T). \end{aligned} \tag{18}$$

Hence the entire right-hand side plant can be viewed as if it was driven by a control signal delayed T_T units of time and has an internal state $x_d(t)$. This state is estimated using a Kalman filter. The time forward observer now generates a predicted state vector $x_p(t)$ (corresponding to the current input $u_m(t)$), from the delayed state vector $x_d(t)$ (which corresponds to the delayed input $u_m(t - T_T)$) according to

$$x_p(t) = e^{AT_T}x_d(t) + \int_{t-T_T}^{t} e^{A(t-\tau)}Bu_m(\tau)\ d\tau \tag{19}$$

and finally the new output is computed as

$$y_p(t) = Cx_p(t) + Du_m(t). \tag{20}$$

The integral term in equation (19) can be computed according to the following state space model

$$\begin{aligned} \dot{z}(t) &= Az(t) + Bu_m(t) \\ g(t) &= z(t) - e^{AT_T}z(t - T_T) \end{aligned} \tag{21}$$

where it can be shown that

$$g(t) = \int_{t-T_T}^{t} e^{A(t-\tau)}Bu_m(\tau)\ d\tau. \tag{22}$$

In this arrangement the predictor does not require any knowledge of the initial conditions and the Kalman filter will eventually converge to the correct internal state of the slave as viewed on the left side of the communication link. Since the internal state of the slave is directly effected when the slave interacts with the environment and the predictor relies on the Kalman filter to estimate the internal state of the salve, no measurements of forces exerted by the remote environment onto the slave are needed.

C. Regulation

For passivity the condition depicted by equation (14) must be met. In other words the predictor must not increase the total energy contained in the returning wave v_m. This condition can be explicitly enforced through the use of a filter, which we refer to as a regulator. First we define $v_t(t)$ as the sum of the returning wave $v_a(t)$ and the prediction $v_p(t)$.

$$v_t(t) = v_a(t) + v_p(t). \tag{23}$$

The goal is to minimize the "distance-to-go" defined as

$$D_{tg}(t) = \int_0^t (v_t(\tau) - v_m(\tau))\ d\tau. \tag{24}$$

For this purpose we define an energy reservoir

$$E_r(t) = \int_0^t ({v_a}^T(\tau)v_a(\tau) - {v_m}^T(\tau)v_m(\tau))\ d\tau \tag{25}$$

which keeps track of the energy extracted by the regulator. The control law which computes v_m in order to drive $D_{tg}(t)$ to zero based on the energy contained in the reservoir is then given by

$$v_m(t) = \alpha\left(1 - e^{-\beta E_r(t)}\right) D_{tg}(t) \tag{26}$$

where α and β are both positive constant tuning parameters. Given that

$$1 > \left(1 - e^{-\beta E_r(t)}\right) > 0 \tag{27}$$

the output $v_m(t)$ and the distance $D_{tg}(t)$ are always of the same sign. At steady state, when the transients have decayed and there is no forced input

$$\lim_{t\to\infty} v_p(t) = \lim_{t\to\infty} v_a(t) = 0 \ \Rightarrow\ \lim_{t\to\infty} v_t(t) = 0. \tag{28}$$

Hence under such condition, from equation (24)

$$\frac{d}{dt}D_{tg}(t) = -v_m(t) \tag{29}$$

given that D_{tg} and v_m are of the same sign, imples that

$$D_{tg}(t) \to 0. \tag{30}$$

If during regulation the energy reserve approaches 0 then

$$\left(1 - e^{-\beta E_r(t)}\right) \to 0 \tag{31}$$

choking the output wave v_m, while simultaneously increasing the energy reserve. At startup it would take a little time for the reservoir to build up, the size of which is governed by β while α determines how fast $D_{tg}(t)$ decays. Choosing α and β to both be positive ensures that the energy reservoir (equation 25) is kept positive.

D. Correcting for the Position Error due to a Varying Delay

It is shown in [8] that the above scheme is stable for a varying delay. Given that position information is encoded in the integral of the wave signal, a varying delay will introduce a non-zero steady state error between the master and the slave. This is because the integral of the wave signals is not preserved as the signals pass through the communication medium. To ensure a zero steady state error we shall add a minor correction $\Delta\hat{u}_m$ to the right moving wave $\hat{u}_m$ (here the 'hat' denotes the original uncorrected signal). Rewriting the expression similar to equation (10), for the position difference as viewed on the left side of the communication line

$$\Delta\theta_2(t) = \theta_m(t) - \theta_{sd}(t - T_L(t)) \tag{32}$$

and expanding yields

$$\begin{aligned}\Delta\theta_2(t) = \tfrac{1}{2}{A_w}^{-1}\textstyle\int_0^t(&\hat{u}_m(\tau) + v_m(\tau) \\ &-u_s(\tau - T_L(\tau)) - v_s(\tau - T_L(\tau)))\ d\tau.\end{aligned} \tag{33}$$

Since we want to correct the right moving wave by adding a corrective term

$$u_m(t) = \hat{u}_m(t) + \Delta\hat{u}_m(t). \tag{34}$$

Assuming $T_L \simeq T_R$, the following approximations can be made

$$u_s(t - T_L(t)) = u_m(t - 2T_L) + \omega_1(t) \tag{35}$$

and

$$v_s(t - T_L(t)) = v_a(t) + \omega_2(t) \tag{36}$$

where ω_1 and ω_2 are assumed to be small perturbations caused by the variations in the delays. Equation (33) can now be written as

$$\begin{aligned}\Delta\theta_2 = &\tfrac{1}{2}{A_w}^{-1}\textstyle\int_{t-2T_L}^t u_m(\tau)d\tau \\ &+\tfrac{1}{2}{A_w}^{-1}\textstyle\int_0^t(v_m(\tau) - v_a(\tau))\ d\tau \\ &-\tfrac{1}{2}{A_w}^{-1}\textstyle\int_0^t \Delta\hat{u}_m(\tau)d\tau\end{aligned} \tag{37}$$

where the perturbations ω_1 and ω_2 have been absorbed into $\Delta\hat{u}_m$. Under ideal conditions $\Delta\hat{u}_m = 0$. Now define the drift error d as

$$d(t) = expected\ \ \Delta\theta_2 - actual\ \ \Delta\theta_2 \tag{38}$$

which leads to

$$\begin{aligned}d(t) = &\tfrac{1}{2}{A_w}^{-1}\textstyle\int_{t-2T_L}^t u_m(\tau) \\ &+\tfrac{1}{2}{A_w}^{-1}\textstyle\int_0^t(v_m(\tau) - v_a(\tau))\ d\tau \\ &-[\theta_m(t) - \theta_{sd}(t - T_L(t))].\end{aligned} \tag{39}$$

However the drift $d(t)$ is also given by

$$d(t) = \frac{1}{2}{A_w}^{-1}\int_0^t \Delta\hat{u}_m(\tau)d\tau. \tag{40}$$

The goal now is to drive the drift $d(t)$ to zero. For this purpose lets define a second energy reservoir which keeps track of how much energy is dissipated in the slave system, as

$$E_d(t) = \int_0^t (\hat{u}_m^T\hat{u}_m - {v_m}^T v_m)\ d\tau. \tag{41}$$

Then let the correction term be computed as

$$\Delta\hat{u}_m = -\gamma(1 - e^{-\delta E_d(t)})A_w d(t) \tag{42}$$

where both γ and δ are positive constants. From equation (40)

$$\frac{d}{dt}d(t) = \frac{1}{2}{A_w}^{-1}\Delta\hat{u}_m. \tag{43}$$

Plugging equation (42) into the above expression yeilds

$$\frac{d}{dt}d(t) = -\frac{1}{2}\gamma(1 - e^{-\delta E_d(t)})d(t). \tag{44}$$

Given that the remote system is dissipative, implies that $E_d(t) \geq 0$ and

$$0 < (1 - e^{-\delta E_d(t)}) < 1. \tag{45}$$

This implies that d(t) and its derivative are always of the opposite signs, hence

$$d(t) \to 0. \tag{46}$$

Note, that the above control law is very similar to that used in the regulator, however the correction term computed is very small given that the variations in the delay are assumed to be small. It should be noted that in this scheme the energy dissipated E_d could temporarily become negative (due to the environment acting on the slave) or large over time. Hence the software should bound the value of E_d between 0 and some small positive number. The overall control scheme is shown in figure 4.

IV. Observer Based Design

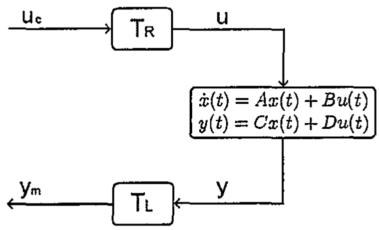

Fig. 5. A linear system with delays in the input and output.

Consider a linear time invariant system, with a delay in the input and a delay in the measured output as shown in figure 5. Here the system itself has the form

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned} \tag{47}$$

where $x(t)$ is an $n \times 1$ state vector, $y(t)$ is an $r \times 1$ output vector and $u(t)$ is an $m \times 1$ input vector. It is assumed that the pair (A, B) is controllable and the pair (A, C) is observable. T_R is the delay in the input and T_L is the delay in the measured output. In other words

$$\begin{array}{l} u(t) = u_c(t - T_R) \\ y_m(t) = y(t - T_L). \end{array} \tag{48}$$

Lets denote the delay in the right direction by $T_R(t)$ and the delay in the left direction by $T_L(t)$. As will be seen that it is only possible to obtain an exact solution when $T_R(t)$ is a constant, while $T_L(t)$ can be varying. Again consider a linear time invariant system with delays in input and output as shown in figure 5. The over all system from the input to the output can now be written as

$$\begin{array}{l} \dot{x}(t) = Ax(t) + Bu_c(t - T_R(t)) \\ y(t) = Cx(t) + Du_c(t - T_R(t)) \\ y_m(t) = y(t - T_L(t)) \end{array} \tag{49}$$

where the output equation can also be written as

$$y_m(t) = Cx(t - T_L(t)) + Du_c(t - T_R(t) - T_L(t)). \tag{50}$$

Let the initial conditions be $x(t_0) = x_o$ and $u(\tau) = u_o(\tau)$ for $-T_R < \tau < 0$. Lets also assume the pair (A, B) is controllable and the pair (A, C) is observable. The solution to a differential equation of the form

$$\dot{x}(t) = Ax(t) + f(t), \qquad x(t_0) = x_0 \tag{51}$$

is given by

$$x(t) = \phi(t, t_0)x(t_0) + \int_{t_0}^{t} \phi(t, \tau)f(t)d\tau \tag{52}$$

where

$$\phi(t, t_0) = e^{A(t-t_0)}. \tag{53}$$

Hence the solution to equation (49) is

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu_c(\tau - T_R(t))\ d\tau \tag{54}$$

which can be written as

$$\begin{array}{l} x(t) = e^{A(t-t_0)}x(t_0) \\ \qquad +e^{-AT_R(t)} \int_{t_0}^{t-T_R(t)} e^{A(t-\tau)}Bu_c(\tau))\ d\tau. \end{array} \tag{55}$$

Similarly

$$\begin{array}{l} x(t + T_R(t)) = e^{A(t+T_R(t)-t_0)}x(t_0) \\ \qquad +e^{-AT_R(t)} \int_{t_0}^{t} e^{A(t+T_R(t)-\tau)}Bu_c(\tau)\ d\tau \end{array} \tag{56}$$

expanding

$$\begin{array}{l} x(t + T_R(t)) = \\ e^{AT_R(t)} \overbrace{\left[e^{A(t-t_0)}x(t_0) + e^{-AT_R(t)} \int_{t_0}^{t-T_R(t)} e^{A(t-\tau)}Bu_c(\tau)d\tau\right]}^{x(t)} \\ \qquad + \int_{t-T_R(t)}^{t} e^{A(t-\tau)}Bu_c(\tau))d\tau \end{array} \tag{57}$$

which can be written as

$$x(t + T_R(t)) = e^{AT_R(t)}x(t) + \int_{t-T_R(t)}^{t} e^{A(t-\tau)}Bu_c(\tau)\ d\tau. \tag{58}$$

Likewise

$$\begin{array}{l} x(t - T_L(t)) = e^{A(t-T_R(t)-t_0)}x(t_0) \\ \qquad +e^{-AT_R(t)} \int_{t_0}^{t-T_R(t)-D_L(t)} e^{A(t-T_L(t)-\tau)}Bu_c(\tau))\ d\tau \end{array} \tag{59}$$

expanding

$$\begin{array}{l} x(t - T_L(t)) = \\ e^{-AT_L(t)} \overbrace{\left[e^{A(t-t_0)}x(t_0) + e^{-AT_R(t)} \int_{t_0}^{t-T_R(t)} e^{A(t-\tau)}Bu_c(\tau)d\tau\right]}^{x(t)} \\ \qquad +e^{-AT_R(t)}e^{-AT_L(t)} \int_{t-T_R(t)}^{t-T_R(t)-T_L(t)} e^{A(t-\tau)}Bu_c(\tau))d\tau \end{array} \tag{60}$$

from which

$$\begin{array}{l} x(t) = e^{AT_L(t)}x(t - T_L(t)) \\ \qquad -e^{-AT_R(t)} \int_{t-T_R(t)}^{t-T_R(t)-T_L(t)} e^{A(t-\tau)}Bu_c(\tau)d\tau. \end{array} \tag{61}$$

Solving for $x(t - T_L(t))$ in terms of $x(t + T_R(t))$ and using equations (58) and (61) yields

$$\begin{array}{l} x(t - T_L(t)) = e^{-AT_R(t)}e^{-AT_L(t)}x(t + T_R(t)) \\ \qquad -e^{-AT_R(t)}e^{-AT_L(t)} \int_{t-T_R(t)-T_L(t)}^{t} e^{A(t-\tau)}Bu_c(\tau)d\tau \end{array} \tag{62}$$

Finally plugging (62) into (50) yields

$$\begin{array}{l} y_m(t) = Ce^{-A(T_R(t)+T_L(t))}x(t + T_R(t)) \\ \qquad -Ce^{-A(T_R(t)+T_L(t))} \int_{t-T_R(t)-T_L(t)}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \\ \qquad\qquad +Du_c(t - T_R(t) - T_L(t)). \end{array} \tag{63}$$

Given that the delay is varying, the differential equation represented by equation (49) can not simply be shifted by $T_R(t)$ units of time into the future. Differentiating the expression for $x(t + T_R(t))$ given by equation (58) yields

$$\begin{array}{l} \dot{x}(t + T_R(t)) = A\dot{T}_R(t)e^{AT_R(t)}x(t) \\ \qquad +e^{AT_R(t)}\dot{x}(t) + \frac{d}{dt}\left[\int_{t-T_R(t)}^{t} e^{A(t-\tau)}Bu_c(\tau)d\tau\right]. \end{array} \tag{64}$$

Using Leibniz rule,

$$\begin{array}{l} \frac{d}{dt} \int_{\alpha(t)}^{\beta(t)} f(x, t)dx = \\ \qquad \int_{\alpha(t)}^{\beta(t)} \frac{\partial}{\partial t}f(x, t)dx + \dot{\beta}(t)f(\beta(t), t) - \dot{\alpha}(t)f(\alpha(t), t) \end{array} \tag{65}$$

the derivative in equation (64) can be computed as

$$\begin{array}{l} \frac{d}{dt}\left[\int_{t-T_R(t)}^{T} e^{A(t-\tau)}Bu_c(\tau)d\tau\right] = \\ \qquad \int_{t-T_R(t)}^{t} Ae^{A(t-\tau)}Bu_c(\tau) + Bu_c(t) \\ \qquad\qquad -e^{AT_R(t)}Bu_c(t - T_R(t)) \\ \qquad\qquad\quad +\dot{T}_R(t)e^{AT_R(t)}Bu_c(t - T_R(t)). \end{array} \tag{66}$$

Plugging the above into (64) yields

$$\begin{aligned}\dot{x}(t+T_R(t)) &= e^{AT_R(t)}Ax(t) \\ &+\int_{t-T_R(t)}^{t} Ae^{A(t-\tau)}Bu_c(\tau)d\tau \\ &+Bu_c(t) + A\dot{T}_R(t)e^{AT_R(t)}x(t) \\ &+\dot{T}_R(t)e^{AT_R(t)}Bu_c(t-T_R(t)).\end{aligned} \tag{67}$$

$\Rightarrow$

$$\begin{aligned}&\dot{x}(t+T_R(t)) = \\ &A\overbrace{\left[e^{AT_R(t)}x(t) + \int_{t-T_R(t)}^{t} e^{A(t-\tau)}Bu_c(\tau)d\tau\right]}^{x(t+T_R(t))} + Bu_c(t) \\ &+A\dot{T}_R(t)e^{AT_R(t)}x(t) + \dot{T}_R(t)e^{AT_R(t)}Bu_c(t-T_R(t)).\end{aligned} \tag{68}$$

Denoting

$$w(t) = A\dot{T}_R(t)e^{AT_R(t)}x(t) + \dot{T}_R(t)e^{AT_R(t)}Bu_c(t-T_R(t)) \tag{69}$$

and assuming that $w(t)$ can be treated as *noise* yields the following linear state-space equation

$$\dot{\overline{x}}(t) = A\overline{x}(t) + Bu_c(t) + w(t) \tag{70}$$

where $x(t+T_R(t))$ is denoted by $\overline{x}(t)$. Lets define $\overline{y}(t)$ as

$$\overline{y}(t) = \overline{C}(t)\overline{x}(t) + Du_c(t) \tag{71}$$

where

$$\overline{C}(t) = Ce^{-A(T_R(t)+T_L(t))}. \tag{72}$$

Using (63) and (71)

$$\begin{aligned}\overline{y}(t) &= y_m(t) \\ &+Ce^{-A(T_R(t)+T_L(t))}\int_{t-T_R(t)-T_L(t)}^{t} e^{A(t-\tau)}Bu_c(\tau)\ d\tau \\ &-Du_c(t-T_R(t)-T_L(t)) + Du_c(t).\end{aligned} \tag{73}$$

For implementation ease the integral in the above equation can be computed as

$$\begin{aligned}\dot{z}(t) &= Az(t) + Bu_m(t) \\ g(t) &= z(t) - e^{AT_T}z(t-T_R(t)-T_L(t))\end{aligned} \tag{74}$$

where it can be shown that

$$g(t) = \int_{t-T_R(t)-T_L(t)}^{t} e^{A(t-\tau)}Bu_c(\tau)\ d\tau. \tag{75}$$

Summarizing the results thus far, a linear time invariant system with delays in the input and the output equations of the form given by equation (49) can be converted into a system of the form

$$\begin{aligned}\dot{\overline{x}}(t) &= A\overline{x}(t) + Bu_c(t) + w(t) \\ \overline{y}(t) &= \overline{C}(t)\overline{x}(t) + Du_c(t).\end{aligned} \tag{76}$$

Note that there are no delays in the system represented by equation (76), which although linear, is time varying. This is because the matrix $\overline{C}(t)$ is no longer constant but is a function of time. Fortunately the state equation is time invariant which can be used to our advantage.

It would be nice if the new system, like the original, was time invariant. Given an estimate of the internal state $\overline{x}(t)$, a new output can be computed according to the following output equation

$$y_n(t) = C\overline{x}_n(t) + Du_c(t) \tag{77}$$

thus eliminating the effect of the time varying matrix $\overline{C}(t)$. For the purpose of this estimation a Kalman filter (see appendix A for more detail) can be used. Here we shall assume that $w(t)$ can be characterized as *white noise* (an assumption necessary for a Kalman filter). Given that the differential equation in the system represented by equation (76) is time invariant, it can be discretized via a Z-transformation. The $\overline{C}(t)$ and D matrices do not have to be discretized and can be directly incorporated into the filter. Hence a generic form of the Kalman filter is perfectly valid where the output equation is time varying. Given that the matrix $\overline{C}(t)$ is a function of the total round trip delay, its value can be determined for every sample time as will be shown in the next chapter. The over all end to end system is given by

$$\begin{aligned}\dot{\overline{x}}(t) &= A\overline{x}(t) + Bu_c(t) \\ y_n(t) &= C\overline{x}_n(t) + Du_c(t).\end{aligned} \tag{78}$$

V. Experimental Setup and Programming Issues

Since the subject of this research is Internet based teleoperation it is important that our experimental apparatus incorporate the actual network in the loop. Given that time delay is introduced only when packets traverse large distances, implies that the master and slave systems have to be separated over a very large geographical region. This is not practical from a research perspective, as we would like to see both the master and slave systems in motion during control implementation, building and testing. In order to overcome this difficulty, the master and slave systems are kept in the same location while the command signal is rebounded from a remote site.

For convenience two different remote sites where chosen, which were located in Metz, France and Tokyo, Japan (more on this later). The local site was Georgia Tech, Atlanta. During development the time delay was simulated in the lab by rebounding the control signals from a third computer, which was programmed to hold each data packet for a specified amount of time. The master manipulator chosen for the experiment was a two degrees of freedom parallel link robot. For kinaesthetic and inertial coupling it is desired that the master and slave robots be identical. Due to the unavailability of two identical systems, the slave robot was simulated in real time on a separate computer and its output was rendered in real time as a 3D image. The overall scheme is shown in figure 6.

The computer controlling the master manipulator is a Windows NT 4.0 based machine. This system is connected to the Internet via a Ethernet card. Since Windows NT is not a real time operating system, it is not suitable for mission critical real time code. For this reason the NT environment is augmented with Hyperkernel software. Code

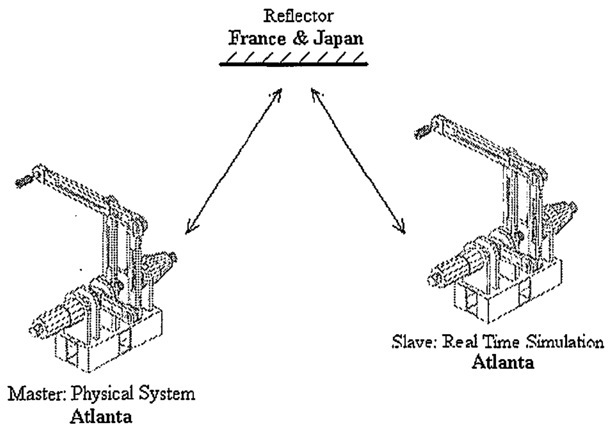

Fig. 6. Experimental setup scheme.

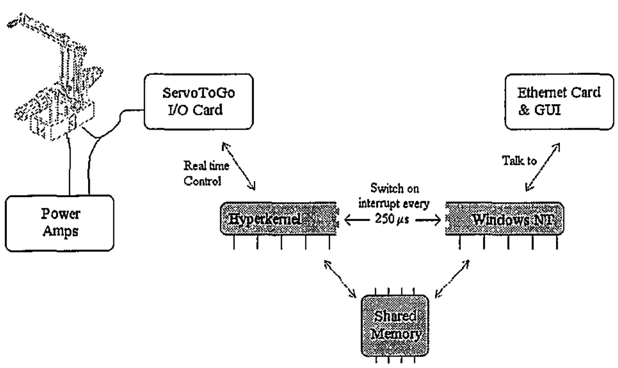

Fig. 7. Master robot and the control software.

executing in the Hyperkernel environment runs in conjunction with the NT environment, where CPU time is evenly shared between NT and the Hyperkernel real time code. Hyperkernel has its own kernel, which has a higher priority than the NT scheduler. Upon receiving an interrupt, the CPU dedicates 100% of its time to the Hyperkernel real time code, on the next interrupt control transfers to Windows NT applications and the process repeats it self. Interrupts can be set to occur at increments ranging from $25\mu s$ to a maximum of $250\mu s$. This way NT applications and real-time code can run concurrently on the same computer.

At any instant only one application can be running, hence communication between the Hyperkernel code and the NT environment is established by writing to a shared memory resource. The over all scheme is shown in figure 7. Hyperkernel has no access to Windows resources such as the screen and Ethernet adapter. For this reason data is exchanged with the slave system by first writing to shared memory which is then exchanged on the Windows NT side over the network. Given that Windows NT can maintain a fairly constant sample time if only one application is running, there does not seem to be any noticeable skips while performing read/write operations between shared memory and the Ethernet adapter.

The user situated at the master computer can view the slave by looking at a real-time 3D rendering of the slave drawn on a second computer. This image is rendered as information becomes available to the master. Meaning that if the delay between the master and slave was $500ms$, then the user would be viewing the slave system as it was $500ms$ ago. Although the image can be render by the master computer, a second computer was used to avoid any non-consistencies (skips) while performing read/write operations between shared memory and the Ethernet adapter (since that part of the code relies on the NT scheduler).

The protocol used to stream the control signal through the network was UDP. This, as shown in [8], provides a consistent sample time yet is non-conformation based. It was observed via trial and error that when data packets were rebounded from France, the control loop could easily be executed at 1KHz. In this case the packet loss was around 1%. However, when packets were reflected from Tokoyo, the control loop had to be run at 250Hz to avoid high packet loss. It is observed that packet loss is not only a function of the distance, but also of the available bandwidth. At small sample times, more data needs to be pumped through the network. Hence if the network does not have the required bandwidth, then with UDP transmission, data is simply lost.

VI. Experimental Results

In the interest of space, only experimental results for the case where the reflector was set up in Tokyo are given. It is not possible to stream data at the same rate between Atlanta and Tokyo as it is between Atlanta and France. Initially, attempts were made to carry out experimental runs with the reflector set up in Japan and the control loop operating at $1KHz$. The result was a choppy response due to high data losses, which in many instances exceeded 90%. The problem of course was bandwidth and not distance. Running the control software at $250Hz$ reduced bandwidth requirements. This alone took care of the data loss problem. Figure 8 shows the closed loop round trip delay between the master and slave with the reflector set up at the Tokyo Institute of Technology, Japan. The average round trip delay was $364.4ms$ with a standard deviation of $26.1ms$ (which is 7.1% of the mean value). The shortest distance along a great circle traversed by a data packet from master to slave and back to the master is over $44000Km$. *This distance exceeds the circumference of the earth*[3]. For this particular result 18762 packets were streamed from the master, of which 18310 were received at the slave. A data loss of about 2.4%.

For completeness results for regular teleoperation are shown in figure 9 with the reflector in Japan. As expected, the system is unstable. From figures 10 and 11 it can be seen that predictive wave-based teleoperation is a significantly more effective than conventional wave-based teleoperation. Even though the fluctuations in delay were much larger, the correction algorithm compensating for position drift derived earlier seems to show good results. It was ob-

[3]The circumference of the earth is approximately 40000 Km

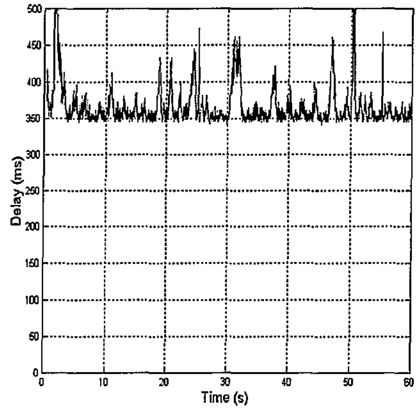

Fig. 8. Closed loop delay between master and slave (data reflected from Tokyo, Japan).

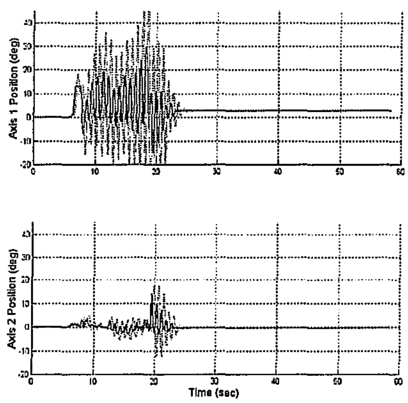

Fig. 9. Regular teleperation teleoperation (data reflected from Tokyo, Japan).

served that at steady state the position of the master and slave systems matched to within $1/1000$ *th* of a radian.

Finally figure 12 shows the results for observer based teleoperation. The results are impressive, since the observer predicts for a varying delay rather than for an averaged delay (as in the predictive wave-based technique). As we shall see shortly, the observer based technique might not be too practical for cases where the slave starts to interact with the environment.

A. Conclusion

During regular teleoperation under no delay (when power variables are transmitted across the communication medium) the user can receive haptic feedback from the remote environment. Consider the slave system touching a wall and the master finally at rest. In that case all velocities are zero, but not the positions. The PD control torque in this case is

$$\tau_{pd} = K_{pd}(\theta_{sd} - \theta_s). \tag{79}$$

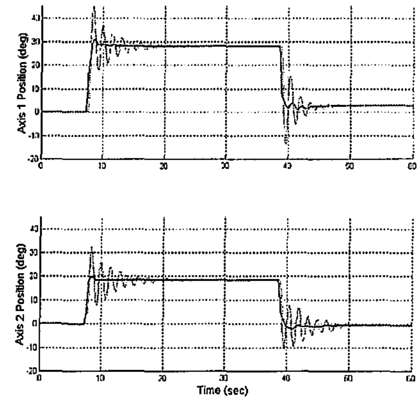

Fig. 10. Wave-based teleoperation (with data reflected from Toyko, Japan).

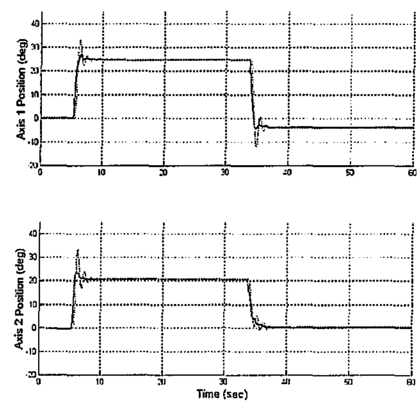

Fig. 11. Wave-based teleoperation with prediction (data reflected from Tokyo, Japan).

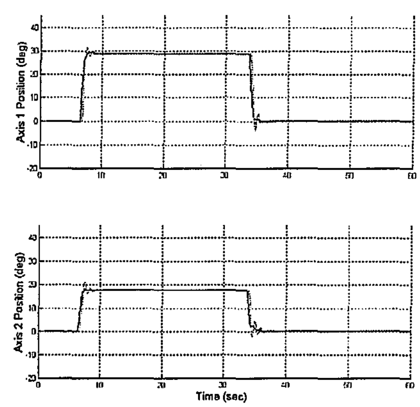

Fig. 12. Observer based teleoperation (data reflected from Tokyo, Japan).

If the user pushes on the master along the direction of the restriction (i.e. the wall) θ_s will not change but θ_{sd} will (since it is the position of master). In this case the commanded torque will change so as to oppose the motion of the master, giving the master a sensation that the slave is encountering a restriction or is in contact. For zero delay this arrangement is perfectly symmetric, hence if the user moves the slave, the master will follow its motion (hence the term bilateral teleoperation). It can be assumed that this is true even for small delays where stability is not an issue. However for larger delays, as we have seen, some sort of delay compensation is needed.

Now lets see what happens when wave variables (without prediction) are used to ensure stability. Utilizing equations (6) and (7) reduces wave-based communication to the following expressions:

$$\begin{aligned} \tau_m(t) &= B_w^{-1} A_w \dot{\theta}_m(t) - B_w^{-1} v_m(t) \\ u_m(t) &= 2A_w \dot{\theta}_m(t) - v_m(t) \end{aligned} \tag{80}$$

and

$$\begin{aligned} \dot{\theta}_{sd}(t) &= A_w^{-1} u_s(t) - A_w^{-1} B_w \tau_s(t) \\ v_s(t) &= u_s(t) - 2B_w \tau_s(t). \end{aligned} \tag{81}$$

At steady state $\dot{\theta}_{sd}(t)$ and $\dot{\theta}_s(t)$ are zero. Hence from equation (80)

$$u_m(t) = -v_m(t). \tag{82}$$

Since nothing is changing at steady state, the following are also true

$$u_s(t) = u_m(t - T_R(t)) = u_m(t) \tag{83}$$

and

$$v_m(t) = v_s(t - T_L(t)) = v_s(t). \tag{84}$$

Plugging equation (82) into (81) yields

$$u_s(t) = B_w \tau_s(t). \tag{85}$$

From equation (80)

$$\tau_m(t) = -B_w^{-1} v_m(t) \tag{86}$$

which together with equations (82), (83) and (84) yields the following critical result

$$\tau_m(t) = \tau_s(t). \tag{87}$$

Hence it can be said that *at steady state, the commanded torques applied to the slave and the master are equal.* Which leads us to conclude that the forces exerted by the environment on the slave are exactly equal to the forces exerted by the user on the master. Implying that the user feels what the slave feels.

At first it may appear that predictive wave-based teleoperation might complicate matters, as the prediction algorithm does not take into account a model of the environment. As already seen, stability is not an issue as the prediction algorithm ensures stability by regulating the return energy. The question however is whether at steady state the forces imposed on the slave are similar to those imposed on the master. From figure 4 it can be seen that the prediction signal is given by

$$v_p(t) = y_p(t) - y_p(t - T_T). \tag{88}$$

Since $y_p(t)$ becomes constant at steady state, implies

$$v_p(t) = 0. \tag{89}$$

This means that at steady state the prediction signal is zero, hence a predictive wave-based teleoperator behaves like a conventional wave-based teleoperator. This (as already stated) allows the commanded torque at the slave and the master to be equal, giving the user haptic feedback of the remote environment.

Finally we look at what happens when teleoperation is conducted with an observer-based controller. Earlier we witnessed remarkable results with the observer. However the performance of the observer is directly related to how accurately a model of the remote system is known. As long as the slave is not interacting with the environment the user can feel a great deal of inertial coupling between the master and slave systems. However if the slave touches a wall, the model of the remote system in the observer dramatically changes. This could yield unpredictable results and might even destabilize the system. Hence with the observer based controller the slave must not pick up heavy objects or engage in hard surface contacts.

Claims made above were all verified experimentally with the aid of a virtual wall in the real time simulation of the slave robot.

Acknowledgments

The help of Professor Fumitoshi Matsuno and Kazuyuki Ito at the Tokyo Institute of Technology with conducting live experiments over the Internet between Atlanta and Tokyo is greatly appreciated. Special thanks also go to Sheryl Raezer and Doug Bakkum with their help in conducting experimental work over the Internet between Atlanta and Metz France.

References

[1] Anderson, Robert J. and Spong Mark W. *Bilateral Control of Teleoperators with Time Delay*, IEEE Transactions On Automatic Control, Vol. 34, No. 5, May 1989, pp 494-501.

[2] Anderson, Robert J., *SMART Class Notes*, Intelligent Systems & Robotics Center, Sandia National Laboratories, July 1994.

[3] Brady, Kevin. *Time-Delayed Control of Telerobotic Manipulators*, Ph.D Thesis, Department of Systems Science and Mathematics, Washington University, August 1997.

[4] Brady, Kevin & Tarn, Tzyh-Jong. *Internet-Based Remote Teleoperation*, IEEE International Conference on Robotics & Automation, Leuven, Belgium, May 1998, pp 65-70.

[5] desoer, C.A. & Vidyasagar, M. *Feedback Systems: Input-Output Properties* Academic Press, 1975.

[6] Munir, Saghir. *Internet-Based Teleoperation*, Ph.D Thesis, George W.Woodruff School of Mechanical Engineering, Georgia Institute of Technology, March 2001.

[7] Munir, Saghir and Book, Wayne. *Wave-Based Teleoperation with Prediction*, American Controls Conference, July 2001.

[8] Munir, Saghir and Book, Wayne. *Internet Based Teleoperation using Wave Variables with Prediction*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Como, Italy, July 2001

[9] Niemeyer, Gunter & Slotine, Jean-Jacques E., *Stable Adaptive Teleoperation*, IEEE Transactions On Automatic Control, Vol. 16, No. 1, Jan 1991, pp 152-162.

[10] Niemeyer, Gunter. *Using Wave Variables in Time Delayed Force Reflecting Teleoperation*, Ph.D Thesis, Dept of Aeronautics and Astronautics, Massachusetts Institute Of Technology, Sep 1996.

[11] Niemeyer, Gunter & Slotine, Jean-Jacques E., *Designing Force Reflecting Teleoperators with Large Time Delays to Appear as Virtual Tools*, IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp 2212-2218.

[12] Niemeyer, Gunter & Slotine, Jean-Jacques E., *Using Wave Variables for System Analysis and Robot Control*, IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp 1619-1625.

[13] Niemeyer, Gunter & Slotine, Jean-Jacques E., *Towards Force-Reflecting Teleoperation Over the Internet*, IEEE International Conference on Robotics & Automation, Leuven, Belgium, May 1998, pp 1909-1915.

[14] Oboe, Roberto & Fiorini, Paolo. *A Design and Control Environment for Internet-Based Telerobotics*, International Journal of Robotics Research, Vol. 17, No. 4, April 1998, pp433-499.

[15] Overstreet, W.Jamahl & Tzes, Anthony. *An Internet-Based Real-Time Control Engineering Laboratory*, IEEE Control Systems Magazine, October 1999, pp 19-34.

[16] smith, O.J.M. *Closer Control of Loops with Dead Time*, Chemical Engineering Progress, Vol.53, No 5, May 1957, pp 217-219.

[17] tarn, Tzyh-Jong & Brady, Kevin. *A Framework for the Control of Time-Delayed Telerobotic Systems*, IFAC Robot Control, SYROCO, Nantes, France, 1997, pp 599-604.

[18] Taylor, Kenneth & Dalton, Barney. *Inernet Robots: A New Robotics Niche*, IEEE Robotics and Automation Magazine, September 1999, pp 27-34.

[19] Watanabe, Keiji & Ito, Masami. *An Observer for Linear Feedback Control Laws of Multivariable Systems with Multiple Delays in Controls and Outputs* , Systems & Control Letters, Vol 1, No. 1, July 1981, pp 54-59.